

How to Verify a Quantum Computation

Anne Broadbent*

Abstract

We give a new theoretical solution to a leading-edge experimental challenge, namely to the verification of quantum computations in the regime of high computational complexity. Our results are given in the language of quantum interactive proof systems. Specifically, we show that any language in BQP has a quantum interactive proof system with a polynomial-time classical verifier (who can also prepare random single-qubit pure states), and a quantum polynomial-time prover. Here, soundness is unconditional—*i.e.* it holds even for computationally unbounded provers. Compared to prior work achieving similar results, our technique does not require the encoding of the input or of the computation; instead, we rely on encryption of the input (together with a method to perform computations on encrypted inputs), and show that the random choice between three types of input (defining a *computational run*, versus two types of *test runs*) suffices. Because the overhead is very low for each run (it is linear in the size of the circuit), this shows that verification could be achieved at minimal cost compared to performing the computation. As a proof technique, we use a reduction to an entanglement-based protocol; this is the first time this technique has been used in the context of verification of quantum computations, and it enables a relatively straightforward analysis.

1 Introduction

Feynman [Fey82] was the first to point out that quantum computers, if built, would be able to perform quantum simulations (*i.e.* to compute the predictions of quantum mechanics; which is widely believed to be classically intractable). But this immediately begs the question: if the output of a quantum computation cannot be predicted, how do we know that it is correct? Conventional wisdom would tell us that we can rely on testing *parts* (or scaled-down versions) of a quantum computer—conclusive results would then extrapolate to the larger system. But this is somewhat unsatisfactory, since we may not rule out the hypothesis that, at a large scale, quantum computers behave unexpectedly. A different approach to the verification of a quantum computation would be to construct a number of quantum computers based on different technologies (*e.g.* with ionic, photonic, superconducting and/or solid state systems), and to accept the computed predictions if the experimental results agree. Again, this is still somewhat unsatisfactory, as a positive outcome does not confirm the correctness of the output, but instead confirms that the various large-scale devices behave similarly on the given instances.

This problem, though theoretical in nature [AV14], is already appearing as a major experimental challenge. One of the outstanding applications for the verification of quantum systems is in quantum chemistry, where the current state-of-the-art is that the inability to verify quantum simulations is much more the norm than the exception [GH05]. Any theoretical advance in this area could have dramatic consequences on applications of quantum chemistry simulations, including the potential

*Department of Mathematics and Statistics, University of Ottawa, Ontario, Canada. Email: abroadbe@uottawa.ca.

to revolutionize drug discovery. Another case where experimental techniques are reaching the limits of classical verifiability is in the Boson Sampling problem [AA11], where the process of verification has been raised as a fundamental objection to the viability of experiments [GKAE13] (fortunately, these claims are refuted [AA14], and progress was made in the experimental verification [SVB⁺14]).

As mere classical probabilistic polynomial-time¹ individuals, we appear to be in an impasse: how can we validate the output of a quantum computation?² For some problems of interest in quantum computing (such as factoring and search), a claimed solution can be efficiently verified by a classical computer. However, current techniques do not give us such an efficient verification procedure for the *hardest* problems that can be solved by quantum computers (such problems are known as BQP-complete, and include the problem of approximating the Jones polynomial [AJL06]). Here, we propose a solution based on *interaction*, viewing an experiment not in the traditional, static, predict-and-verify framework, but as an interaction between an experimentalist and a quantum device. In the context of theoretical computer science, it has been established for quite some time that interaction between a probabilistic polynomial-time *verifier* and a computationally unbounded *prover* allows the verification of a class of problems *much* wider than what static proofs allow³.

Interactive proof systems traditionally model the prover as being all-powerful (*i.e.* computationally unbounded)⁴. For our purposes, we restrict the prover to being a “realistic” quantum device, *i.e.* we model the prover as a quantum polynomial-time machine. Our approach equates the verifier with a classical polynomial-time machine, augmented with *extremely* rudimentary quantum operations, namely of being able to prepare random single pure-state qubits (chosen among a specific set, see Section 3). Our verifier does not require any quantum memory or quantum processing power. Without loss of generality, the random quantum bits can be sent in the first round, the balance of the interaction and verifier’s computation being classical. Formally, we present our results in terms of an *interactive proof system*, showing that in our model, it is possible to devise a *quantum prover interactive proof system* for all problems solvable (with bounded error) in quantum polynomial time.

1.1 Related Work

The complexity class QPIP, corresponding to quantum prover interactive proof systems, was originally defined by Aharonov, Ben-Or and Eban [ABE10], who, using techniques from [BOCG⁺06], showed that $\text{BQP} = \text{QPIP}$ for a verifier with the capacity to perform quantum computations on a constant-sized quantum register (together with polynomial-time classical computation). The main idea of [ABE10] is to encode the input into a *quantum authentication code* [BCG⁺02], and to use interactive techniques for *quantum computing on authenticated data* in order to enable verification of a quantum computation. This result was revisited in light of foundations of physics in [AV14], and the protocol was also shown secure in a scenario of *composition* [BGS13].

In a different line of research, Kashefi and Fitzsimons [FK12] consider a measurement-based approach to the problem, giving a scheme that requires the verifier to prepare only random single qubits: the main idea is to encode the computation into a larger one which includes a verification mechanism, and to execute the resulting computation using blind quantum computing [BFK09]. Thus, success of the encoded computation can be used to deduce the correctness of actual computa-

¹*i.e.* assuming humans can flip coins and execute classical computations that take time polynomial in n to solve on inputs of size n .

²Assuming the widely-held belief that $\text{BQP} \neq \text{BPP}$, *i.e.* that quantum computers are indeed more powerful than classical computers.

³This is the famous $\text{IP} = \text{PSPACE}$ result [LFKN92, Sha92].

⁴Notable exceptions include [CL92, GKR08].

tion. A small-scale version of this protocol was implemented in quantum optics [BFKW13]. Further work by Kapourniotis, Dunjko and Kashefi [KDK15] shows how to combine the [ABE10] and [FK12] protocols in order to reduce the quantum communication overhead; Kashefi and Wallden [KW15] also show how to reduce the overhead of [FK12].

To the best of our knowledge, the proof techniques in these prior works appear as sketches only, or are cumbersome. In particular, the approach that uses quantum authentication codes [ABE10] is based on [BOCG⁺06]. However, the full proof of security for [BOCG⁺06] never appeared. Although [ABE10] makes significant progress towards closing this gap, it provides only a sketch of how the soundness is established in the interactive case. A full proof of soundness for [ABE10] follows from [BGS13], however the proof is very elaborate and phrased in terms of a rather different cryptographic task (called “quantum one-time programs”). In terms of the measurement-based approach, note that a proposed protocol for verification in [BFK09] was deemed incomplete [FK12], but any gaps were addressed in [FK12]. In this case, however, the protocol (and proof) are very elaborate, and to the best of our knowledge, remain unpublished.

In sharp contrast to these approaches, Reichardt, Unger and Vazirani [RUV13] show that it is possible to make the verifier *completely* classical, as long as we postulate *two* non-communicating entangled provers (this could be enforced, for instance, by space-like separation such that communication between the provers would be forbidden by the limit on the speed of light). The main technique used is a *rigidity theorem* which, provided that the provers pass a certain number of tests, gives the verifier a tight classical control on the quantum provers.

1.2 Contributions

Our main contributions are a new, simple quantum prover interactive proof system for BQP, with a verifier whose quantum power is limited to the random preparation of single-qubit pure states, together with a new proof technique:

New protocol. All prior approaches to the verification of quantum computations required some type of encoding (either of the input or of the computation). In contrast, our protocol achieves soundness via the verifier’s random choice of different types of runs. This is a typical construction in interactive proofs, and in some sense it is surprising that it is used here for the first time in the context of verifying quantum computations. According to the new protocol, the overhead required for verification can be reduced to repetition of a very simple protocol (with overhead at most linear compared to performing the original computation), and thus may lead to implementations sooner than expected (in general, it is much easier to repeat an experiment using different initial settings, than to run a single, more complex experiment!).

New proof technique. In order to prove soundness, we use the proof technique of a reduction to an “entanglement-based” protocol. This proof technique originates from Shor and Preskill [SP00] and has been used in a number of quantum cryptographic scenarios, *e.g.* [DFSS05, DFPR14, FBS⁺14]. This is the first time that this technique is used in the context of the verification of quantum computations; we show how the technique provides a much-needed succinct and convincing method to prove soundness. In particular, it allows us to reduce the analysis of an interactive protocol to the analysis of a non-interactive one, and to formally delay the verifier’s choice of run until *after* the interaction with the prover.

Furthermore, this work unifies the two distinct approaches given above, (one based on quantum authentication codes and the other on measurement-based quantum computing). Indeed, one can view our protocol as performing a very basic type of quantum computing on authenticated

data [BGS13]; with hidden gates being executed via a computation-by-teleportation process [GC99] that is reminiscent of measurement-based quantum computation, and thus of blind quantum computation [BFK09].

On the conceptual front, this work focuses on the *simplest possible* way to achieve a quantum prover interactive proof system. Via this process, we have further emphasized links between various concepts:

1. **A link between input privacy and verification.** Prior results [ABE10, BFK09, BGS13, FK12] all happened to provide both *privacy* of a quantum computation and its *verification* (one notable exception being the recent [FH15]). Here, we make this link explicit, essentially starting from input privacy and constructing a verifiable scheme (this was also done, to a certain extent in [BFK09, FK12]).
2. **A link between fault-tolerant quantum computation and cryptography.** Prior results [BOCG⁺06, ABE10, BGS13] used constructions inspired by fault-tolerant quantum computation. Here, we make the link even more explicit by using single-qubit gate gadgets that are adaptations of the gate gadgets used in fault-tolerant quantum computation. Furthermore, our results also emphasize how the ubiquitous technique of “tracking the Pauli frame” from fault-tolerant quantum computation can be re-phrased in terms of keeping track of an encryption key.
3. **A link between entanglement and parallelization.** It is known that entanglement can reduce the number of rounds in quantum interactive proof systems [KKMV08]; a consequence of our entanglement-based protocol is that we can parallelize our interactive proof system to a single round, as long as we are willing to allow the prover to share entanglement with the verifier, and to perform adaptive measurements.

1.3 Overview of Techniques

The main idea for our quantum prover interactive proof system is that the verifier chooses randomly to interact with the prover in one of three runs. Among these runs, one is the *computation* run, while the two others are *test* runs. In an honest interaction, the output of the computation run is the result (a single bit) of evaluating the given quantum circuit. The test runs are used to detect a deviating prover; there are two types of test runs: an *X-test* and a *Z-test*. Intuitively (and formally proved in Section 7.1), we see that the prover cannot distinguish between all three runs. Thus, his strategy must be invariant over the different runs. It should be clear now how this work links *input privacy* with verification: by varying the input to the computation, the verifier differentiates between test and computation runs; by input privacy, however, the prover cannot identify the type of run and thus any deviation from the prescribed protocol has a chance of being detected.

In more details, the runs have the following properties (from the point of view of the verifier)

- **Computation run.** In a computation run, the prover executes the target circuit on input $|0\rangle^{\otimes n}$.
- **X-test run.** In an X-test run, the prover executes the identity circuit on input $|0\rangle^{\otimes n}$. At the end of the computation, the verifier verifies that the result is 0. This test also contains internal checks for cheating within the protocol.
- **Z-test run.** In a Z-test run, the prover executes the identity circuit on input $|+\rangle^{\otimes n}$. This test run is used only as an internal check for cheating within the protocol.

In order for the prover to execute the above computations without being able to distinguish between the runs, we use a technique inspired by *quantum computing on encrypted data (QCED)* [FBS⁺14, Bro15]: the input qubits are encrypted with a random Pauli, as are auxiliary qubits that are used to drive the computation.

Viewing the target computation as a sequence of gates in the universal gateset $\{X, Z, H, \text{CNOT}, T\}$ (see Section 2.1 for notation), the task we face is, in the computation run, to perform these logical gates on encrypted quantum data. Furthermore, the X - and Z -test runs should (up to an encryption key), leave the quantum wires in the $|0\rangle^{\otimes n}$ or $|+\rangle^{\otimes n}$ state, respectively. Performing Pauli gates in this fashion is straightforward, as this can be done by adjusting the encryption key (in the computation run only). As we show in Section 4.2, the CNOT gate can be executed directly (since it does not have any effect on the wires for the test runs). The T -gate (Section 4.3) is performed using a construction (“gate gadget”) inspired both by QCED and fault-tolerant quantum computation [BMP⁺00] (see also [BJ15]); the T -gate gadget involves the use of an auxiliary qubit and classical interaction. The H is performed thanks to an identity involving the H and P (Section 4.4). Note that P can be accomplished as T^2 .

In order to prove soundness, we consider any general deviation of the prover, and show that such deviation can be mapped to an attack on the measured wires only, corresponding to an honest run of the protocol (without loss of generality, we can also delay all measurements until the end of the protocol). Furthermore, because the computation is performed on encrypted data, by the *Pauli twirl* [DCEL09], this attack can be described as a convex combination of Pauli attacks on the measured qubits. Since all measurements are performed in the computational basis, Z attacks are obliterated, and thus the only family of attacks of concern consists in X - and Y -gates applied to various measured qubits; these act as bit flips on the corresponding classical output. We show that the combined effect of test runs is to detect *all* such attacks; this allows us to bound the probability that the verifier accepts a *no*-instance. Since only X and Y attacks require detection, one may wonder why we use also a Z -test run. The answer to this question lies in the implementation of the H -gate: while its net effect is to apply the identity in the test runs, its internal workings temporally *swap* the roles of the X - and Z -test runs: thus the Z -test runs are also used to detect X and Y errors.

Finally, some words on showing indistinguishability between the test and computation runs. This is done by showing that the verifier can delay her choice of run (computation, X - or Z -test) until *after* the interaction with the prover is complete. This is accomplished via an entanglement-based protocol, where the verifier’s messages to the prover consist in only half-EPR pairs, as well as classical random bits. These messages are identical in both the test and computation runs; as the verifier decides on the type of run only *after* having interacted with the prover. Depending on this choice, the verifier performs measurements on the system returned by the prover, resulting in the desired effect.

1.4 Open Problems

The main outstanding open problem is the verifiability of a quantum computation with a *single*, classical verifier, interacting with a quantum polynomial-time prover. In this context, we mention a few observations:

- If the prover is unbounded, there exists a quantum interactive proof system for BQP, since $\text{QIP} = \text{PSPACE}$ [JJUW10].
- If $P = \text{BQP}$, there is a trivial quantum interactive proof system.
- One possible approach would be to relax the definition to require only *computational* soundness (following the lines of Brassard, Chaum and Crépeau [BCC88], this would lead to a quantum interactive *argument*). This approach seems promising, especially if we consider a computational assumption that is *post-quantum* secure. If, via its interaction with the prover, a classical verifier accepts, then we can conclude that either the verifier performed the correct computation *or* the prover has broken the computational assumption.

1.5 Organization

The remainder of this paper is organized as follows. Section 2 presents some preliminary notation and background. Section 3 defines quantum prover interactive proofs and states our main theorem. Section 4 describes the interactive proof system, for which we show completeness (Section 6), and soundness (Section 7).

2 Preliminaries

2.1 Notation

We assume the reader is familiar with the basics of quantum information [NC00]. We use the following well-known qubit gates $X : |j\rangle \mapsto |j \oplus 1\rangle$, $Z : |j\rangle \mapsto (-1)^j |j\rangle$, Hadamard $H : |j\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^j |1\rangle)$, phase gate $P : |j\rangle \mapsto i^j |j\rangle$ and $\pi/8$ rotation $T : |j\rangle \mapsto e^{(i\pi/4)^j} |j\rangle$ and the two-qubit controlled-not $CNOT : |j\rangle|k\rangle \mapsto |j\rangle|j \oplus k\rangle$. Let $Y = iXZ$. We denote by \mathbb{P}_n the n -qubit Pauli group, where a $P \in \mathbb{P}_n$ is given by $P = P_1 \otimes P_2 \otimes \cdots \otimes P_n$ where $P_i \in \{I, X, Y, Z\}$.

2.2 Quantum Encryption and the Pauli Twirl

The quantum one-time pad encryption maps a single-qubit system ρ to $\frac{1}{4} \sum_{a,b \in \{0,1\}} X^a Z^b \rho Z^b X^a = \frac{I}{2}$; its generalization to n -qubit systems is straightforward [AMTW00]. Here, we take (a, b) to be the classical private *encryption key*. Clearly, this scheme provides information-theoretic security, while allowing decryption, given knowledge of the key. A useful observation is that if we have an *a priori* knowledge of the quantum operator ρ , then it may not be necessary to encrypt it with a full quantum one-time pad (*e.g.* if the state corresponds to a pure state of the form $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$, it can be encrypted with a random Z), although there is no loss of generality in encrypting it with the full random Pauli. We use the two interpretations interchangeably.

Consider for a moment the classical one-time pad (that encrypts a plaintext message by XORing it with a random bit-string of the same length). It is intuitively clear that if an adversary (who does not know the encryption key) has access to the ciphertext only, and is allowed to modify it, then the effect of any adversarial attack (after decryption) is to probabilistically introduce bit flips in target locations. The quantum analogue of this is given by the *Pauli twirl* [DCEL09]:

Lemma 1. (*Pauli Twirl*) Let P, P' be n -qubit Pauli operators. Then:

$$\frac{1}{|\mathbb{P}_n|} \sum_{Q \in \mathbb{P}_n} Q^* P Q \rho Q^* P'^* Q = \begin{cases} 0, & P \neq P' \\ P \rho P^*, & \text{otherwise.} \end{cases}$$

3 Definitions and Statement of Results

Interactive proof systems were introduced by Babai [Babai85] and Goldwasser, Micali, and Rackoff [GMR89]. An interactive proof system consists of an interaction between a computationally unbounded prover and a computationally bounded probabilistic verifier. For a language L and a string x , the prover attempts to convince the verifier that $x \in L$, while the verifier tries to determine the validity of this “proof”. Thus, a language L is said to have an interactive proof system if there exists a polynomial-time verifier V such that:

- (Completeness) if $x \in L$, there exists a prover (called an honest prover) such that the verifier accepts with probability $p \geq 2/3$;

- (Soundness) if $x \notin L$, no prover can convince V to accept with probability $p \geq 1/3$.

The class of languages having interactive proof systems is denoted IP .

Watrous [Wat03] defined QIP as the quantum analogue of IP , *i.e.* as the class of languages having a *quantum* interactive proof system, which consists in a quantum interaction between a computationally unbounded quantum prover and a computationally bounded quantum verifier, with the analogous completeness and soundness conditions as given above.

For our results, we are interested in the scenario of a polynomial-time prover (in the honest case), as well as an *almost-classical* verifier; that is, a verifier with the power to generate random qubits as specified by a parameter \mathcal{S} (Definition 1). Furthermore, as a technicality, instead of considering languages, we consider promise problems: a promise problem $\Pi = (\Pi_Y, \Pi_N)$, is a pair of disjoint sets of strings, corresponding to YES and NO instances, respectively. For a formal treatment of the model (which we specialize here to our scenario), see [Wat03].

Definition 1. Let $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_\ell\}$ where $\mathcal{S}_i = \{\rho_1, \dots, \rho_{\ell_i}\}$ ($i = 1, \dots, \ell$) is a set of density operators. A \mathcal{S} -Quantum Prover Interactive Proof System for a promise problem $\Pi = (\Pi_Y, \Pi_N)$ is an interactive proof system with a verifier V that runs in classical probabilistic polynomial time, augmented with the capacity to randomly generate states in each of $\mathcal{S}_1, \dots, \mathcal{S}_\ell$ (upon generation, these states are immediately sent to the prover, with its index $i \in \{1, \dots, \ell\}$ known to the verifier). The verifier V interacts with a prover P such that:

- (Completeness) if $x \in \Pi_Y$, there exists a quantum polynomial-time prover (called an honest prover) such that the verifier accepts with probability $p \geq 2/3$;
- (Soundness) if $x \in \Pi_N$, no prover (even unbounded) can convince V to accept with probability $p \geq 1/3$.

The class of promise problems having an \mathcal{S} -quantum interactive proof systems is denoted $\text{QPIP}_{\mathcal{S}}$. Note that by standard amplification, the class $\text{QPIP}_{\mathcal{S}}$ is unchanged if we replace the completeness parameter c and soundness parameter s by any values, as long as $c - s > \frac{1}{\text{poly}(n)}$.

Comparing our definition of $\text{QPIP}_{\mathcal{S}}$ to the class of quantum prover interactive proof systems (QPIP) as given in [ABE10], we note that we have made some modifications and clarifications, namely that the verifier in $\text{QPIP}_{\mathcal{S}}$ does not have any quantum memory and does not perform any gates (QPIP allows a verifier that stores and operates on a quantum register of a constant number, c , of qubits), and that soundness holds against unbounded provers.

Finally, we use the same BQP-complete promise problem as [ABE10]: (the canonical BQP-complete problem.)

Definition 2. The promise problem *Q-CIRCUIT* consists of a quantum circuit made of a sequence of gates, $U = U_T, \dots, U_1$ acting on n input qubits (we take these circuits to be given in the universal gateset $\{\text{X}, \text{Z}, \text{H}, \text{CNOT}, \text{T}\}$). Let $p(U) = \|\lvert 0 \rangle \langle 0 \rvert \otimes \mathbb{I}_{n-1} U \lvert 0^n \rangle\|^2$ be the probability of observing “0” as a results of a computational basis measurement of the n^{th} output qubit, obtained by evaluating U on input $\lvert 0^n \rangle$.

Then define $\text{Q-CIRCUIT} = \{\text{Q-CIRCUIT}_{\text{YES}}, \text{Q-CIRCUIT}_{\text{NO}}\}$ with:

$$\text{Q-CIRCUIT}_{\text{YES}} : p(U) \geq 2/3 \quad (1)$$

$$\text{Q-CIRCUIT}_{\text{NO}} : p(U) \leq 1/3 \quad (2)$$

We can now formally state our main theorem.

Theorem 1 (Main Theorem). Let $\mathcal{S} = \{\lvert 0 \rangle, \lvert 1 \rangle\}, \lvert + \rangle, \lvert - \rangle\}, \{\text{P} \lvert + \rangle, \text{P} \lvert - \rangle\}, \{\text{T} \lvert + \rangle, \text{T} \lvert - \rangle, \text{PT} \lvert + \rangle, \text{PT} \lvert - \rangle\}$. Then $\text{BQP} = \text{QPIP}_{\mathcal{S}}$.

4 Quantum Prover Interactive Proof System

In order to prove Theorem 1, we give an interactive proof system (see **Interactive Proof System 1**). This protocol uses the various gate gadgets as described in Sections 4.1–4.4. Completeness is studied in Section 6 and soundness is proved in Section 7.

Interactive Proof System 1 Verifiable quantum computation with trusted auxiliary states

Let \mathcal{C} be given as an n -qubit quantum circuit in the universal gateset X, Z, CNOT, H, T .

1. The verifier randomly chooses to execute one of the following three runs (but does not inform the prover of this choice).

A. Computation Run

- A.1. The verifier encrypts input $|0\rangle^{\otimes n}$ and sends the input to P .
- A.2. The verifier sends auxiliary qubits required for the T -gate gadgets for the computation run as given in Sections 4.4 and 4.3.
- A.3. For each gate G in \mathcal{C} : X, Z and CNOT are performed without any auxiliary qubits or interaction as given in Sections 4.1 and 4.2, while the H - and T -gate gadgets are performed using the auxiliary qubits from Step A.2. and the interaction as given in Sections 4.4 and 4.3, respectively.
- A.4. P measures the output qubit and returns the result to V .
- A.5. V decrypts the answer; let the result be c_{comp} . V accepts if $c_{\text{comp}} = 0$; otherwise reject.

B. X-test Run

- B.1. The verifier encrypts input $|0\rangle^{\otimes n}$ and sends the input to P .
- B.2. The verifier sends auxiliary qubits required for the T -gate gadgets for the X-test run as given in Sections 4.4 and 4.3.
- B.3. For each gate G in \mathcal{C} : X, Z and CNOT are performed without any auxiliary qubits or interaction as given in Sections 4.1 and 4.2, while the H - and T -gate gadgets are performed using the auxiliary qubits from Step B.2. and the interaction as given in Sections 4.4 and 4.3, respectively.
- B.4. P measures the output qubit and returns the result to V .
- B.5. V decrypts the answer; let the result be c_{test} . V accepts if no errors were detected in step B.3. and if $c_{\text{test}} = 0$; otherwise reject.

C. Z-test Run

- C.1. The verifier encrypts input $|+\rangle^{\otimes n}$ and sends the input to P .
 - C.2. The verifier sends auxiliary qubits required for the T -gate gadgets for the Z-test run as given in Sections 4.4 and 4.3.
 - C.3. For each gate G in \mathcal{C} : X, Z and CNOT are performed without any auxiliary qubits or interaction as given in Sections 4.1 and 4.2, while the H - and T -gate gadgets are performed using the auxiliary qubits from Step C.2. and the interaction as given in Sections 4.4 and 4.3, respectively.
 - C.4. P measures the output qubit and returns the result to V .
 - C.5. V disregards the output. V accepts if no errors were detected in step C.3.; otherwise reject.
-

4.1 X- and Z-gate gadget

In order to apply an X on a qubit register i encrypted with key (a_i, b_i) , the verifier updates the key according to $a_i \leftarrow a_i \oplus 1$ (b_i is unchanged). In order to apply an Z on a qubit register i encrypted

with key (a_i, b_i) , the verifier updates the key according to $b_i \leftarrow b_i \oplus 1$ (a_i is unchanged). This operation is performed only in the computation run.

4.2 CNOT-gate gadget

In order to apply a CNOT gate on the encrypted registers (say with register i being the control and register j the target), the prover simply applies the CNOT gate on the respective registers. The verifier updates the encryption keys according to $a_i \leftarrow a_i; b_i \leftarrow b_i \oplus b_j; a_j \leftarrow a_i \oplus a_j$; and $b_j \leftarrow b_j$. We mention that $\text{CNOT}(|0\rangle|0\rangle) = |0\rangle|0\rangle$ and $\text{CNOT}(|+\rangle|+\rangle) = |+\rangle|+\rangle$; thus in the X- and Z-test runs, the underlying data is unchanged.

4.3 T-gate gadget

Here, we show how the T is performed on encrypted data. This is accomplished using an auxiliary qubits, as well as classical interaction. For the computation run, we use a combination of a protocol inspired from [FBS⁺14, Bro15], as well as fault-tolerant quantum computation [BMP⁺00] (see also [BJ15]). This is given in Figure 1. In the case of an X and Z test runs, as usual, we want the identity map to be applied. This is done as in Figures 2 and 3, respectively. Correctness of Figure 1 is proven in Section 5. Note that we show in Section 6.1 that the set of auxiliary quantum states required by the verifier can be reduced via a simple re-labelling, in order to match the resources requires in Theorem 1.

Also, note that in this work, we have slightly sacrificed efficiency for clarity in the proof, namely that we could have defined a P-gadget using one simple auxiliary qubit instead of two auxiliary qubits that are used by implementing the P as T². Furthermore, we suspect that the P^y gate is unnecessary in Figure 1 and thus that we can simplify the set \mathcal{S} (however, the proof in this case appears to be more elaborate, so once more we choose clarity of the proof over efficiency).

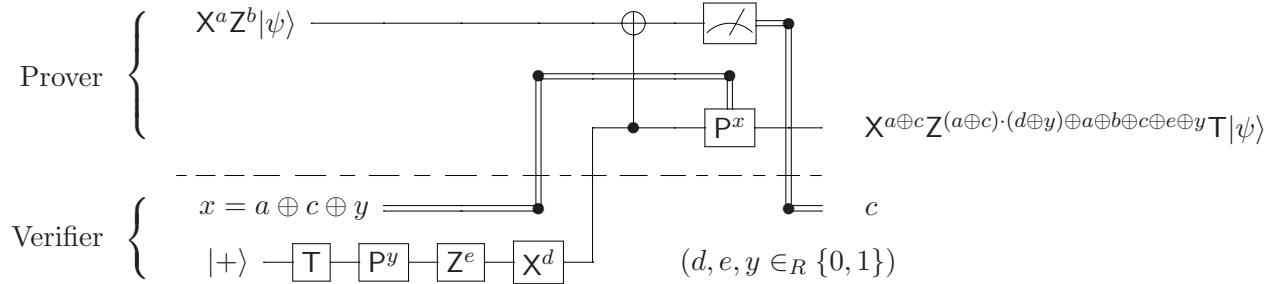


Figure 1: T-gate gadget for a computation run. Here, an auxiliary qubit is prepared by the verifier in the state $X^d Z^e P^y T |+\rangle$ and sent to the prover. The prover performs a CNOT between the auxiliary register and the data register; and then measures the data register. Given the measurement result, c , the verifier sends a classical message, $x = a \oplus c \oplus y$ to the prover, who applies the conditional gate P^x to the remaining register (which we now re-label as the data register). The verifier's key update rule is given in the figure.

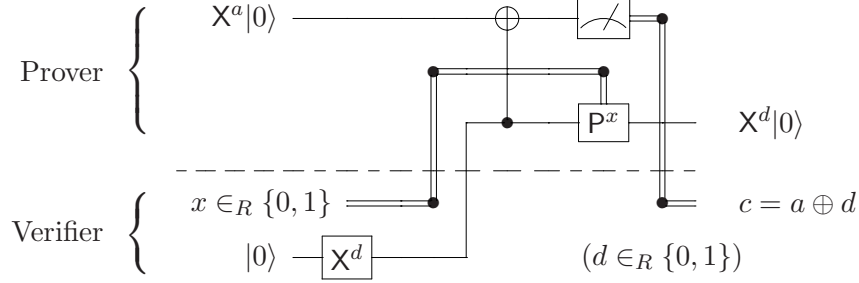


Figure 2: T-gate gadget for an X-gate test run. The goal here is to mimic the interaction established in Figure 1, but to perform the identity operation on the input state $|0\rangle$ (up to encryptions). Here, we include an additional *verification* that $c = a \oplus d$.

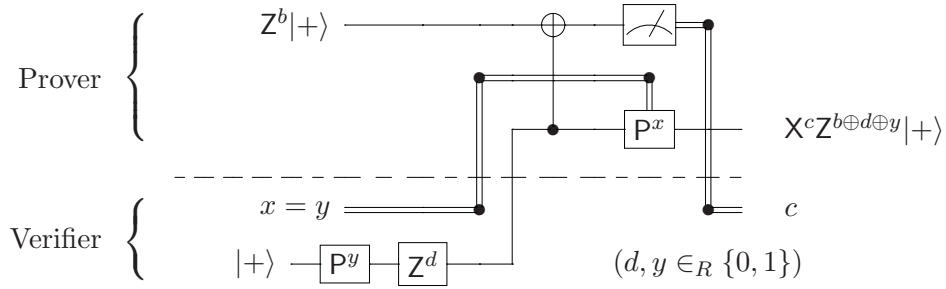


Figure 3: T-gate gadget for a Z-gate test run. The goal here is to mimic the interaction established in Figure 1, but to perform the identity operation on the input state $|+\rangle$ (up to encryptions)

4.4 H-gate gadget

Performing a H gate has the effect of swapping between the X- and Z-test runs (as well as swapping the role of the X and Z encryption keys in the computation run). While this is alright if done in isolation, it does not work if the H-gate is performed as part of a larger computation (for instance, a CNOT-gate could no longer be performed as given above as the inputs would not, in general, be of the form $|0\rangle|0\rangle$ (for the X-test run) or $|+\rangle|+\rangle$ (for the Z-test run). Our solution is to use the following two identities:

$$\text{HPHPHPH} = \text{H} \quad (3)$$

$$\text{HHHH} = \mathbb{I}. \quad (4)$$

Thus we build the gadget so that the prover starts by applying an H at the start. By doing this, we swap the roles of the X- and Z-tests. For the following P, we apply twice the gadgets from Section 4.3 (taking in to account the swapped role for the test runs). The result is that a P is applied in the computation run, while the identity is applied in the test runs. Now an H is applied, which reverts the roles of the X- and Z-tests. We apply the P again. Continuing in this fashion, the effect is:

1. In the computation run (using twice the gadget of Figure 1 for each P-gate), the effect is to apply H on the input qubit (by Equation 3).
2. In the X-test run (using (twice each time) the gadgets of Figures 3, 2, 3 for the first, second and third P-gate), the effect is to apply the identity.
3. In the Z-test run (using (twice each time) gadgets of Figure 2, 3, 2 for the first, second and third P-gate), the effect is to apply the identity.

5 Correctness of the T-gate protocol

We give below a step-by-step proof of the correctness of the T-gate protocol as given in Figure 1 (Section 4.3). The basic building block is the circuit identity for an X-teleportation from [ZDC00]. Also of relevance to this work are the techniques developed by Childs, Leung, and Nielsen [CLN05] to manipulate circuits that produce an output that is correct *up to known Pauli corrections*.

We will make use of the following identities which all hold up to an irrelevant global phase: $XZ = ZX$, $PZ = ZP$, $PX = XZP$, $TZ = ZT$, $TX = XZPT$, $P^2 = Z$ and $P^{a \oplus b} = Z^{a \cdot b} P^{a+b}$ (for $a, b \in \{0, 1\}$).

1. We start with the “X-teleportation” of [ZDC00], which is easy to verify (Figure 4).

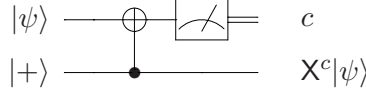


Figure 4: Circuit identity: “X-teleportation”.

2. Then we substitute the input $X^a Z^b |\psi\rangle$ for the top wire. We add the gate sequence $T, P^y, Z^e, X^d, P^{a \oplus c \oplus y}$ to the output (Figure 5). By Figure 4, the outcome is given by $P^{a \oplus c \oplus y} X^d Z^e P^y T X^{a \oplus c} Z^b |\psi\rangle$.

We apply the identities given above to simplify this to a Pauli correction (on T) only:

$$P^{a \oplus c \oplus y} X^d Z^e P^y T X^{a \oplus c} Z^b = P^{a \oplus c \oplus y} X^d Z^e P^y X^{a \oplus c} P^{a \oplus c} Z^{b \oplus a \oplus c} T \quad (5)$$

$$= P^{a \oplus c \oplus y} X^d Z^e P^y P^{a \oplus c} X^{a \oplus c} Z^b T \quad (6)$$

$$= P^{a \oplus c \oplus y} P^y X^d Z^{d \cdot y \oplus e} P^{a \oplus c} X^{a \oplus c} Z^b T \quad (7)$$

$$= P^{a \oplus c \oplus y} P^y P^{a \oplus c} X^d Z^{d \cdot (a \oplus c)} Z^{d \cdot y \oplus e} X^{a \oplus c} Z^b T \quad (8)$$

$$= P^{(a \oplus c) \oplus y} P^y P^{a \oplus c} X^{a \oplus c \oplus d} Z^{d(a \oplus c \oplus y) \oplus b \oplus e} T \quad (9)$$

$$= Z^{y \cdot (a \oplus c)} P^{a \oplus c} P^y P^y P^{a \oplus c} X^{a \oplus c \oplus d} Z^{d(a \oplus c \oplus y) \oplus b \oplus e} T \quad (10)$$

$$= X^{a \oplus c \oplus d} Z^{(a \oplus c \oplus d) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus d \oplus e \oplus y} T \quad (11)$$

$$= X^{a \oplus c'} Z^{(a \oplus c') \cdot (d \oplus y) \oplus a \oplus b \oplus c' \oplus e \oplus y} T \quad (12)$$

Where above, we let $c' \leftarrow c \oplus d$.

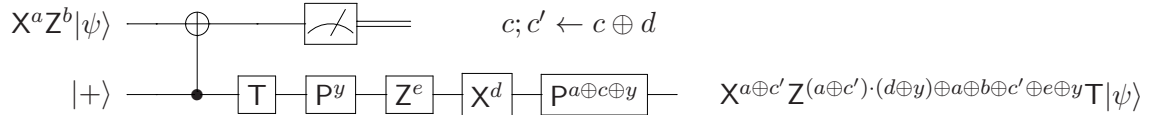


Figure 5: Circuit identity: adding gates to the “X-teleportation”. The output is computed in Equations 5–12.

3. Next, we note that, because diagonal gates commute with control, the circuit of Figure 5 is equivalent to the one in Figure 6.

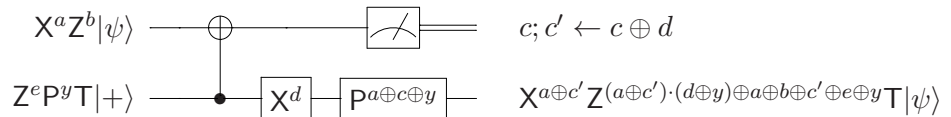


Figure 6: Circuit equivalent to Figure 5, since diagonal gates commute with control.

4. We note that the X^d on the bottom wire *after* the CNOT can be moved to the bottom wire *before* the CNOT, as long as we add an X^d to the top wire after the CNOT. (Figure 7).

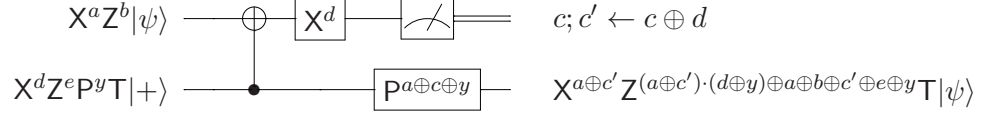


Figure 7: We move the X^d to the input of the bottom wire.

5. Finally, since $c' = c \oplus d$, yet the measurement result c undergoes an X^d , these two operations cancel out, and we obtain the final circuit as in Figure 8.

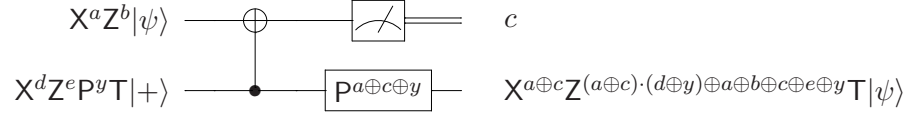


Figure 8: We move the X^d to the input of the bottom wire.

We note that a more direct proof of correctness for Figure 8 is possible, but that our intermediate Figure 7 is crucial in the proof of soundness.

6 Completeness

Suppose \mathcal{C} is a yes-instance. Suppose P follows the protocol honestly. Then we have the following:

1. In the case of a computation run, the output bit, c_{comp} has the same distribution as the output bit of $C(|0^n\rangle)$, thus V accepts with probability at least $2/3$.
2. In the case of an X-test run and in the case of a Z-test run (by the identities and observations from the previous sections), V accepts with probability 1.

Given that each run happens with probability $\frac{1}{3}$, we get that V accepts with probability at least $\frac{2}{3} + \frac{1}{3} \cdot \frac{2}{3} = \frac{8}{9}$.

6.1 Auxiliary qubits for the T-gate gadget

In the protocol for the T-gate gadget (Figure 1), we assume the verifier can produce auxiliary qubits of the form $X^d Z^e P^y T|+\rangle$. We now show that this is equivalent to requiring the prover to generate auxiliary qubits of the form $Z^e P^y T|+\rangle$, as claimed in Theorem 1. This is true because, up to global phase:

$$X^d Z^e P^y T|+\rangle = Z^{e \oplus d} P^{y \oplus d} T|+\rangle \quad (13)$$

The above can be seen easily since, up to global phase, $XT|+\rangle = ZPT|+\rangle$, and $XP = ZPX$, thus for the case $d = 1$, we get:

$$XZ^e P^y T|+\rangle = Z^e X P^y T|+\rangle \quad (14)$$

$$= Z^e Z^y P^y XT|+\rangle \quad (15)$$

$$= Z^{e \oplus y} P^y ZPT|+\rangle \quad (16)$$

$$= Z^{e \oplus y \oplus 1} P^{y+1} T|+\rangle \quad (17)$$

$$= Z^{e \oplus y \oplus 1} Z^y P^{y \oplus 1} T|+\rangle \quad (18)$$

$$= Z^{e \oplus 1} P^{y \oplus 1} T|+\rangle \quad (19)$$

Thus, the verifier chooses a classical x uniformly at random, and if $x = 1$, the verifier re-labels the auxiliary qubits according to Equation 13.

7 Soundness

As discussed in Section 1.3, the main idea to prove soundness is to analyze an entanglement-based version of the Interactive Proof System 1. We present the EPR-based version (Section 7.1), and show that, for any prover, the interactive proof systems are indistinguishable (and thus the completeness and soundness parameters are the same). Then, we analyze a general deviating prover P^* in the EPR-based version and show how to simplify an attack (Section 7.2). We then analyze the case of a test run (Section 7.4) and of a computation run (Sections 7.5). This completes the proof of our main theorem (Theorem 1).

An interesting consequence of the analysis in this section is that it implies that, if we are willing to have the prover and the verifier share entanglement, then the protocol reduces to a single round (however, in this case, the work of the verifier becomes more important; one can wonder if the verifier is still “almost-classical”). Another interesting observation is that sequential repetition is not required (parallel repetition suffices), due to the fact that the analysis makes use of the Pauli twirl (see Section 7.2), which would also be applicable to the scenario of parallel repetition.

7.1 EPR-based protocol

In this version of the quantum prover interactive proof system (**Interactive Proof System 2**), *all* quantum inputs sent by the verifier are half-EPR pairs, and *all* classical messages sent by the verifier are random bits. The actions related to choosing between test and computation runs are done *after* the interaction with the server. For the T-gate, this can be done as shown in Figures 9, 10 and 11.

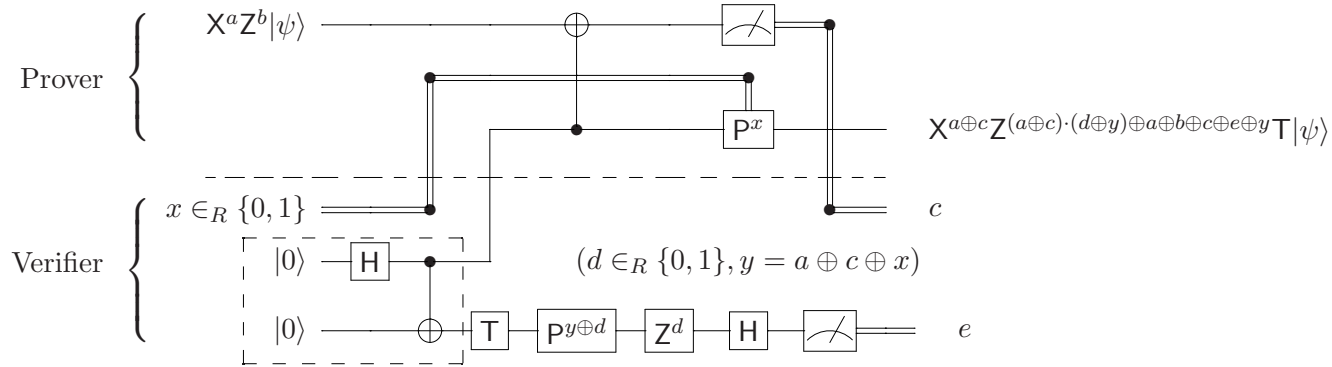


Figure 9: Entanglement-based protocol for a T-gate (computation run). This protocol performs the same computation as the protocol in Figure 1, as can be seen via Equation ???. The output is obtained from the output of Figure 1 by using $y = a \oplus c \oplus x$. The circuit in the dashed box prepares an EPR-pair.

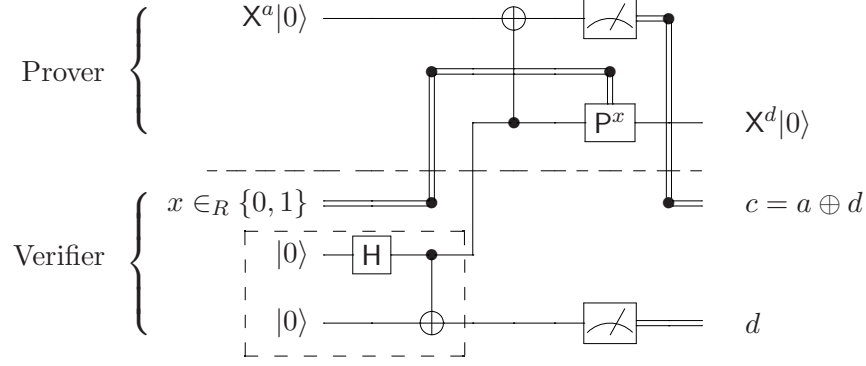


Figure 10: Entanglement-based protocol for an X-test run). This protocol performs the same computation as the protocol in Figure 2. The circuit in the dashed box prepares an EPR-pair. As in Figure 2, we include an additional *verification* that $c = a \oplus d$.

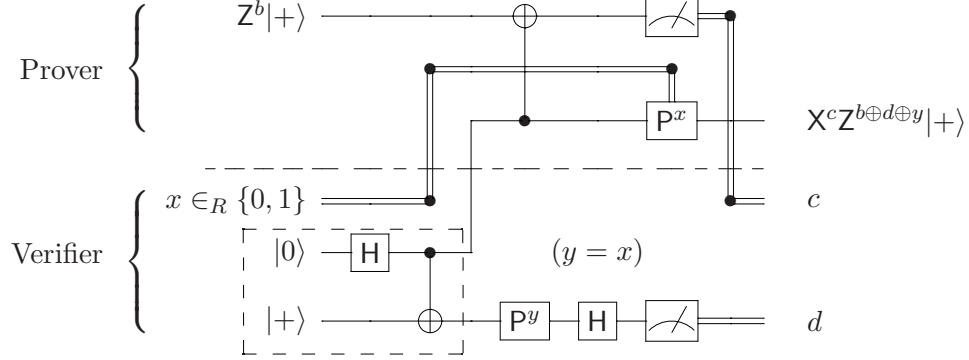


Figure 11: Entanglement-based protocol for an X-test run (Z-test run). This protocol performs the same computation as the protocol in Figures 3. The circuit in the dashed box prepares an EPR-pair.

We therefore define V_{EPR} as a verifier that delays all choices until after the Prover has returned all messages (*i.e.* as the verifier in Interactive Proof System 2). Clearly, no prover P^* can distinguish between an interaction with V or V_{EPR} . This is because we can first fix x to be random in the T-gate gadget in Interactive Proof System 1. This determines y , but the operation, P^y , that is conditioned on y can be formally delayed until *after* the interaction without changing the prover's view (via the EPR-picture).

Thus the soundness parameter is the same for both protocols. Furthermore, clearly the completeness parameter is unchanged. From now on, we therefore focus on establishing the soundness parameter for Interactive Proof System 2.

7.2 Simplifying a general attack

We now consider a general deviating prover P^* interacting with V_{EPR} . First, we make a simplifying assumption that P^* does not perform any measurements (and instead returns single qubits to V , who then performs the measurement). Next, we show that, without loss of generality, we can assume that the prover's actions are the honest ones, followed by a general attack (on the measured and traced-out qubits). In order to see this, for a k -round protocol (involving k rounds of classical interaction), define P 's actions at round i by $\Phi_i C_i$, where C_i acts on the qubits used in the computation, as well as the classical bit received in round i , and is the honest application

Interactive Proof System 2 Verifiable quantum computation with trusted auxiliary states- EPR version

Let \mathcal{C} be given as an n -qubit quantum circuit in the universal gateset X, Z, CNOT, H, T .

1. The verifier prepares $|\Phi^+\rangle^{\otimes n}$ and sends half of each pair to the prover. These registers are identified with the *input* registers.
2. For each auxiliary qubit required in the H - and T -gate gadgets, the verifier prepares $|\Phi^+\rangle$ and sends half of each pair to the prover.
3. The prover executes the gate gadgets. The verifier records the classical communication and responds with random classical bits (when required).
4. The prover returns a single bit of output, c to the verifier.
5. The verifier randomly chooses to execute one of the following three runs (but does not inform the prover of this choice).

A. Computation Run

- A.1. Measure the remaining input register halves in the computational basis. Take the initial X -encryption key to be the measurement outcomes (set the Z -key to 0).
- A.2. For each gate G in \mathcal{C} : perform the key updates for the X, Z , CNOT and H gates. For the T gadget, taking into account the classical message received and sent in Step 2, perform the measurement and key update rules for the T -gadget (Figure 9).
- A.3. V decrypts the output bit c ; let the result be c_{comp} . V accepts if $c_{\text{comp}} = 0$; otherwise reject.

B. X-test Run

- B.1. Measure the remaining input register halves in the computational basis. Take the initial X -encryption key to be the measurement outcomes (set the Z -key to 0).
- B.2. For each gate G in \mathcal{C} : perform the key updates for the X, Z , CNOT and H gates. For the T gadget, taking into account the classical message received and sent in Step 2, perform the measurement, key update rules and tests for the T -gadget (Figure 10).
- B.3. V decrypts the output bit c ; let the result be c_{comp} . V accepts if $c_{\text{comp}} = 0$ *and* if no errors were detected in step B.2.; otherwise reject.

C. Z-test Run

- C.1. Measure the remaining input register halves in the Hadamard basis. Take the initial Z -encryption key to be the measurement outcomes (set the X -key to 0).
 - C.2. For each gate G in \mathcal{C} : perform the key updates for the X, Z , CNOT and H gates. For the T gadget, taking into account the classical message received and sent in Step 2, perform the measurement, key update rules and tests for the T -gadget (Figure 11).
 - C.3. V accepts if no errors were detected in step C.2.; otherwise reject.
-

of the prover's circuit, while Φ_i is a general deviating map acting on the classical bit received in round i , the output registers of C_i as well as a private memory register. (Recall that there are no measurements at this point — V does the measurement).

Thus the actions of a general prover P^* can be described as:

$$\Phi_r C_r \dots \Phi_1 C_1 \Phi_0 C_0. \quad (20)$$

Since C_0, \dots, C_r are unitary, we can re-write Equation 20 as:

$$\Phi_r C_r \dots \Phi_3 C_3 \Phi_2 C_2 \Phi_1 C_1 \Phi_0 C_0 = \Phi_r C_r \dots \Phi_3 C_3 \Phi_2 C_2 (\Phi_1 C_1 \Phi_0 C_1^*) C_1 C_0 \quad (21)$$

$$= \Phi_r C_r \dots \Phi_3 C_3 (\Phi_2 C_2 \Phi_1 C_1 \Phi_0 C_1^* C_2^*) C_2 C_1 C_0 \quad (22)$$

$$= \Phi_r C_r \dots (\Phi_3 C_3 \Phi_2 C_2 \Phi_1 C_1 \Phi_0 C_1^* C_2^* C_3^*) C_3 C_2 C_1 C_0 \quad (23)$$

$$= (\Phi_r C_r \dots \Phi_3 C_3 \Phi_2 C_2 \Phi_1 C_1 \Phi_0 C_1^* C_2^* C_3^* \dots C_r^*) C_r \dots C_3 C_2 C_1 C_0 \quad (24)$$

Thus, by denoting a general attack by $\Phi = \Phi_r C_r \dots \Phi_3 C_3 \Phi_2 C_2 \Phi_1 C_1 \Phi_0 C_1^* C_2^* C_3^* \dots C_r^*$, and the map corresponding to the honest prover as $C = C_r \dots C_3 C_2 C_1 C_0$, we get that without loss of generality, we can assume that the prover's actions are the honest ones, followed by a general attack:

$$\Phi C \quad (25)$$

Taking E_k to be Kraus terms associated with Φ , and supposing a total of m qubit registers are involved, we get that the system after interaction with P^* , where the initial state is $|\Phi^+\rangle\langle\Phi^+|^{\otimes m} = \frac{1}{2^m} \sum_{i,j=0}^{2^m-1} |ii\rangle\langle jj|$ (here, we include the classical random bits, as they are uniformly random and therefore we can represent them as maximally entangled states), and where the verifier has not yet performed the gates and measurements, can be described as:

$$\frac{1}{2^m} \sum_k \sum_{i,j} (I \otimes E_k C) |ii\rangle\langle jj| (I \otimes C^* E_k^*) \quad (26)$$

For a fixed k , we write E_k and E_k^* in the Pauli basis: $E_k = \sum_Q \alpha_Q Q$ and $E_k^* = \sum_{Q'} \alpha_{Q'}^* Q'$ (to simplify notation, we assume throughout that Q, Q' ranges over \mathbb{P}_m). For such an E_k , we now consider the probability of acceptance, depending on the type of run (the following analysis can be applied to each operator in the Kraus decomposition).

7.3 Conventions and definitions

In addition to the convention of representing an attack as in Equation 26, in the following Sections 7.4–7.5, we use the following conventions:

1. The circuit C that we consider is already “compiled” in terms of the H gates as in Section 4.4 (the identity $H = HPHPH$ is already applied).
2. The input prepared by the verifier (in step A.1., B.1. or C.1. of Protocol 2) given as $\sum_P P|\psi\rangle\langle\psi|P^*$ (an encrypted version of $|\psi\rangle\langle\psi|$, where $|\psi\rangle$ is either $|0\rangle$ or $|+\rangle$).
3. The number of T-gate gadgets is t (each such gadget uses two auxiliary qubits—one representing an auxiliary quantum bit, and one representing a classical bit x), and the number of qubits in the computation is n , thus we have $m = 2t + n$.
4. In the T-gate gadget, the auxiliary wire is swapped with the measured wire immediately before the measurement; this way, we may picture that only auxiliary qubits are measured as part of the computation, and that the data registers for the input $|\psi\rangle$ represent the computation wires throughout.

5. Given the system as in Equation 26, we suppose that the first T-gate gadget uses the first EPR pair as auxiliary quantum bit, and the second EPR pair as a qubit representing the classical bit (and so on for the following T-gadgets). The last n EPR pairs are the data qubits, and we suppose that at the end of the protocol, the last data qubit is the one that is measured, representing the output.
6. Normalization constants are omitted when they are clear from context.

Finally, we define *benign* and *non-benign* Pauli attacks, based on their effect on the protocol: as we will see, benign attacks have no effect on the acceptance probability (because all qubits are either traced-out or measured in the computational basis). However, non-benign attacks may influence the acceptance probability.

Definition 3. For a fixed Pauli $P \in \mathbb{P}_m$, we call it benign if $P \in B_{t,n}$, where $B_{t,n}$ is the set of Paulis acting on $m = 2t + n$ qubits, such that the measured qubits in the protocol are acted on only by a gate in $\{I, Z\}$. Using the above conventions, this means that $B_{t,n} = \{\{I, Z\}^t \mathbb{P}_{n-1} \{I, Z\}\}$. A Pauli P is called non-benign if at least one measured qubit in the protocol is acted on only by a gate in $\{X, Y\}$. In analogy to the set of benign Paulis, we denote the set of non-benign Paulis acting on $m = 2t + n$ qubits as $B'_{t,n}$.

7.4 In the case of a test run

Based on the preliminaries of Sections 7.2 and 7.3, we now bound the probability of acceptance of the test runs, by describing the effect of the attack on the entire system, and considering which attacks are detected by the test runs (essentially, we show in Lemma 2 that all non-benign attacks are detected by one of the test runs).

Lemma 2. Consider the Interactive Proof System 2 for a circuit C on n qubits and with t T-gate gadgets, with attack $E_k = \sum_Q \alpha_Q Q$. Let $B'_{t,n}$ be the set of non-benign attacks. Then with the following probability, one of the test runs will reject:

$$\sum_{Q \in B'_{t,n}} |\alpha_Q|^2, \quad (27)$$

Proof. In the case of a test run, the effect of V_{EPR} 's operations, (consisting in gates and measurements) is to prepare an encrypted version of an input state $|\phi\rangle|\psi\rangle$ which is $|\psi\rangle = |0\rangle^n$ for the X-test run or $|\psi\rangle = |+\rangle^n$ in the Z-test run, together with $|\phi\rangle$ consisting in auxiliary qubits, which are a tensor product of systems $|0\rangle$ and $\mathbb{P}^y|+\rangle$ (the choice of which state depending on the number of H-gates that have been applied so far, as per the H-gate construction in Section 4.4). Furthermore, we can assume that the classical random bits are obtained by a computational basis measurement on V_{EPR} 's portion of the maximally entangled states and included as encryptions of $|0\rangle$ in $|\phi\rangle$.

Thus V 's operations can be seen as the preparation of $|\phi\rangle|\psi\rangle$, up to some uniformly distributed random Pauli operator P on the entire space of $|\phi\rangle|\psi\rangle$. Representing C as the honest prover's operations, the system is thus described as:

$$\sum_P E_k C P (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) P^* C^* E_k^*, \quad (28)$$

where P ranges over all Pauli operators $P \in \mathbb{P}_m$. Recall that we have assumed that the verifier performs the measurement in place of the prover; we can further delay this measurement until *after* the verifier has applied the decryption operation. In this case, the decryption key \tilde{P} is computed

such that $CP = \tilde{P}C$. Thus after decryption, the system is:

$$\sum_P \tilde{P}^* E_k C P (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) P^* C^* E_k^* \tilde{P} = \sum_P \tilde{P}^* E_k \tilde{P} C (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) C^* \tilde{P}^* E_k^* \tilde{P}. \quad (29)$$

The map $P \mapsto \tilde{P}$ is a bijection, hence we can rewrite Equation 29 as:

$$\sum_P P^* E_k P C (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) C^* P^* E_k^* P. \quad (30)$$

Using the Pauli decomposition for E_k , we can write this as:

$$\sum_{Q, Q'} \sum_P \alpha_Q \alpha_{Q'}^* P^* Q P C (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) C^* P^* Q' P. \quad (31)$$

By Lemma 1 (Pauli twirl), Equation 31 simplifies to:

$$\sum_Q |\alpha_Q|^2 Q C (|\phi\rangle\langle\phi| \otimes |\psi\rangle\langle\psi|) C^* Q^*. \quad (32)$$

Let $Q \in B'_{t,n}$. Then we claim that at least one of the test runs (an X- or Z-test run) rejects. This is clear because each time a qubit is measured, it will be in the state $|0\rangle$ in exactly one of the X- or Z-test runs (recall, we have stripped away the encryption; also, according to the honest prover's actions C , if an even number of H gates are applied, the X-test run will have auxiliary qubit $|0\rangle$; if an odd number of H gates are applied, then the Z-test run will have auxiliary qubit $|0\rangle$). Thus, the verifier always detects a non-benign attack in one of the test runs. Thus with probability $\sum_{Q \in B'_{t,n}} |\alpha_Q|^2$, one of the test runs rejects. \square

7.5 In the case of a computation run

Still using the notation of Sections 7.2 and 7.3, we now analyze soundness in the case of a computation run. First, we determine the effect of a bit flip on the measured qubit in the T-gate gadget (Section 7.5.1), then we do the analysis for the case that the computation run consists in a single T-gadget (Section 7.5.2); this is extended to full generality in Section 7.5.3.

7.5.1 Effect of a bit flip on a measured qubit

In Lemma 3, we establish the effect of a bit flip on the measured qubit in the T-gate gadget.

Lemma 3. *The error induced by an X-gate on the measured qubit in the T-gate gadget in Figure 9 to introduce an extra $Z^{a \oplus c \oplus d \oplus x} P$ on the output.*

Proof. We determine below the effect of an X-gate on the measured qubit in Figure 1 to be the application of an extra $Z^{d \oplus y} P$ on the output. Since $y = a \oplus c \oplus x$ in Figure 9, this yields the effect of applying $Z^{a \oplus c \oplus d \oplus x} P$ on the output.

An X-gate on the measured qubit in Figure 1 will cause the bottom wire to receive the correction $P^{a \oplus c \oplus y \oplus 1}$ (instead of $P^{a \oplus c \oplus y}$). Since $P^{a \oplus c \oplus y \oplus 1} = P Z^{a \oplus c \oplus y} P^{a \oplus c \oplus y}$, we use and revise the calculation from Equations 5–12, as follows:

$$P^{a \oplus c \oplus y \oplus 1} X^d Z^e P Y T X^{a \oplus c} Z^b = P Z^{a \oplus c \oplus y} P^{a \oplus c \oplus y} X^d Z^e P Y T X^{a \oplus c} Z^b \quad (33)$$

$$= P Z^{a \oplus c \oplus y} X^{a \oplus c \oplus d} Z^{(a \oplus c \oplus d) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus d \oplus e \oplus y} T \quad (34)$$

$$= X^{a \oplus c \oplus d} Z^{(a \oplus c \oplus d) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus d \oplus e \oplus y} Z^{a \oplus c \oplus y} Z^{a \oplus c \oplus d} P T \quad (35)$$

$$= X^{a \oplus c \oplus d} Z^{(a \oplus c \oplus d) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus d \oplus e \oplus y} (Z^{d \oplus y} P) T \quad \square$$

7.5.2 The T-gate protocol under attack

Here, we study the effect of an attack $E_k = \sum_Q \alpha_Q Q$, $E_k^* = \sum_{Q'} \alpha_{Q'}^* Q'$, considering the case of a single-qubit input and of a single T protocol; the general case (Section 7.5.3) follows this analysis closely.

Consider an attack on the protocol as given in Figure 9, and suppose the verifier applies the corrections conditional on c coherently, as well as the decryption coherently (without performing a measurement), but otherwise the verifier performs the measurements on the half-EPR pairs (so that a, b, d, e, x are all classical, uniformly random bits). According to Figure 7, we can see the attacks as being applied before the measurement and decryption on the top wire (this allows the application of the Pauli twirl).

Given that we are analyzing a single T-gate gadget, we note that each Pauli Q, Q' in the attack consists in three single-qubits Paulis: $Q = Q_1 \otimes Q_2 \otimes Q_3$, $Q' = Q'_1 \otimes Q'_2 \otimes Q'_3$. According to our convention, the Paulis Q_1 and Q'_1 act on the measured qubit in Figure 9, the Paulis Q_2 and Q'_2 acts on the traced-out qubit that represents the classical bit x (which we represent as $X^x|0\rangle$) and the Paulis Q_3 and Q'_3 act on the output qubit (also eventually measured).

Let E be the error on the output induced by an X on the top wire in Figure 9; by Lemma 3, $E = Z^{a \oplus c \oplus d \oplus x} P$. For a Pauli $Q \in \mathbb{P}_1$, let $\delta_Q = 0$ if $Q \in \{I, Z\}$ and $\delta_Q = 1$ otherwise. Ignoring the normalization, for a fixed k , the system becomes:

$$\begin{aligned} \sum_{Q, Q'} \alpha_Q \alpha_{Q'}^* \sum_{a, b, c, d, e, x \in \{0, 1\}} & X^d Q_1 X^d |c\rangle \langle c| X^d Q'_1 X^d \otimes Q_2 X^x |0\rangle \langle 0| X^x Q'_2 \otimes \\ & X^{a \oplus c} Z^{(a \oplus c) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus e \oplus y} Q_3 X^{a \oplus c} Z^{(a \oplus c) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus e \oplus y} E^{\delta_{Q_1}} T |0\rangle \\ & \langle 0| T E^{\delta_{Q'_1}} X^{a \oplus c} Z^{(a \oplus c) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus e \oplus y} (Q'_3) X^{a \oplus c} Z^{(a \oplus c) \cdot (d \oplus y) \oplus a \oplus b \oplus c \oplus e \oplus y} \end{aligned} \quad (36)$$

Since a and b appear only on the last register, we can re-write the above as:

$$\begin{aligned} \sum_{Q, Q'} \alpha_Q \alpha_{Q'}^* \sum_{c, d, e, x \in \{0, 1\}} & X^d Q_1 X^d |c\rangle \langle c| X^d Q'_1 X^d \otimes Q_2 X^x |0\rangle \langle 0| X^x Q'_2 \otimes \\ & \sum_{a, b} X^a Z^b X^{f(a, b, c, d, x)} Z^{g(a, b, c, d, x)} Q_3 X^{f(a, b, c, d, x)} Z^{g(a, b, c, d, x)} X^a Z^b E^{\delta_{Q_1}} T |0\rangle \\ & \langle 0| T E^{\delta_{Q'_1}} X^a Z^b X^{f(a, b, c, d, x)} Z^{g(a, b, c, d, x)} (Q'_3) X^{f(a, b, c, d, x)} Z^{g(a, b, c, d, x)} X^a Z^b, \end{aligned} \quad (37)$$

Where f and g are functions of a, b, c, d, x that can be derived from the previous equation. We apply the Pauli twirl (Lemma 1) to the last register. The cross-terms cancel out, and the decryption is applied. Letting $\alpha_Q = \alpha_{Q_1} \alpha_{Q_2} \alpha_{Q_3}$ (and similarly for $\alpha_{Q'}$), we get:

$$\begin{aligned} \sum_{Q_1, Q'_1, Q_2, Q'_2, Q_3} \alpha_{Q_1} \alpha_{Q_2} \alpha_{Q'_1}^* \alpha_{Q'_2}^* |\alpha_{Q_3}|^2 \sum_{c, d, x \in \{0, 1\}} & X^d Q_1 X^d |c\rangle \langle c| X^d Q'_1 X^d \otimes Q_2 X^x |0\rangle \langle 0| X^x Q'_2 \otimes \\ & Q_3 E^{\delta_{Q_1}} T |0\rangle \langle 0| T E^{\delta_{Q'_1}} Q_3 \end{aligned} \quad (38)$$

Since the second register is traced out, we can assume it is decrypted before the trace-out; we

can also move the quantifier over x to the second register:

$$\sum_{Q_1, Q'_1, Q_2, Q'_2, Q_3} \alpha_{Q_1} \alpha_{Q_2} \alpha_{Q'_1}^* \alpha_{Q'_2}^* |\alpha_{Q_3}|^2 \sum_{c, d \in \{0,1\}} X^d Q_1 X^d |c\rangle \langle c| X^d Q'_1 X^d \otimes \sum_{x \in \{0,1\}} X^x Q_2 X^x |0\rangle \langle 0| X^x Q'_2 X^x \otimes Q_3 E^{\delta_{Q_1}} T |0\rangle \langle 0| T E^{\delta_{Q'_1}} Q_3 \quad (39)$$

Next, we apply the Pauli twirl to the first and second registers to get:

$$\sum_{Q_1, Q_2, Q_3} |\alpha_{Q_1}|^2 |\alpha_{Q_2}|^2 |\alpha_{Q_3}|^2 \sum_{c \in \{0,1\}} Q_1 |c\rangle \langle c| Q_1 \otimes Q_2 |0\rangle \langle 0| Q_2 \otimes Q_3 E^{\delta_{Q_1}} T |0\rangle \langle 0| T E^{\delta_{Q_1}} Q_3 \quad (40)$$

From the above, it is clear that the outcome after measuring the first and last register (and tracing out the second register) is the correct computation (*i.e.* a computational basis measurement on $T|0\rangle$) if $Q = Q_1 \otimes Q_2 \otimes Q_3$ is benign. Thus the outcome is correct with probability at least $\sum_{Q \in B_{1,1}} |\alpha_Q|^2$.

7.5.3 General analysis for a computation run

We now bound the acceptance probability in the case of a computation run. In analogy to the case of the test runs, we show that an attack E_k can be broken down into a convex combination of Pauli attacks; this is possible essentially because, from the provers's point of view, the system is encrypted. More formally, we follow the lines of the case analyzed in Section 7.5.2, and write down an expression for the system after the verifier's coherent operations. Thanks to the Pauli twirl, we can simplify this expression considerably. Then, we show how to bound the acceptance probability of benign attacks (no such bound is possible for non-benign attacks), which gives us the desired result.

Lemma 4. *Consider the Interactive proof system 2 for a circuit C on n qubits and with t T-gate gadgets, with attack $E_k = \sum_Q \alpha_Q Q$. Let $B_{t,n}$ be the set of benign attacks, and $B'_{t,n}$ be the set of non-benign attacks. Then the probability of acceptance for a computation run is at most:*

$$p \sum_{Q \in B_{t,n}} |\alpha_Q|^2 + \sum_{Q \in B'_{t,n}} |\alpha_Q|^2 \quad (41)$$

Where $p = \|(|0\rangle\langle 0| \otimes \mathbb{I}_{n-1})C|0^n\rangle\|^2$ is the probability that we observe “0” as a results of a computational basis measurement of the n^{th} output qubit, obtained by evaluating C on input $|0^n\rangle$.

Proof. The proof is a generalization of the techniques developed in Section 7.5.2. Following the conventions of Section 7.3, we ignore normalization constants. Let E be the error on the output induced by an X on the top wire in Figure 9; by Lemma 3, $E = Z^{a \oplus c \oplus d \oplus x} P$. For a Pauli $Q \in \mathbb{P}_1$, let $\delta_Q = 0$ if $Q \in \{I, Z\}$ and $\delta_Q = 1$ otherwise. We let $Q = P_1 \otimes Q_1 \otimes P_2 \otimes Q_2 \otimes \dots \otimes P_t \otimes Q_t \otimes R$, with $P_i, Q_i \in \mathbb{P}_1$ and $R \in \mathbb{P}_n$, and we decompose the target circuit C as $C = C_t T \dots T C_1 T C_0$, where each C_i is a Clifford group circuit.

As Section 7.5.2, we use the identity from Figure 7. For a fixed k , the system after the attack, and the verifier's coherent corrections is thus:

$$\begin{aligned}
& \sum_{Q, Q'} \alpha_Q \alpha_{Q'}^* \left(\sum_{c_1, d_1, e_1, x_1 \in \{0,1\}} X^{d_1} P_1 X^{d_1} |c_1\rangle \langle c_1| X^{d_1} P'_1 X^{d_1} \otimes Q_1 X^{x_1} |0\rangle \langle 0| X^{x_1} Q'_1 \otimes \dots \otimes \right. \\
& \quad \sum_{c_t, d_t, e_t, x_t \in \{0,1\}} X^{d_t} P_t X^{d_t} |c_t\rangle \langle c_t| X^{d_t} P'_t X^{d_t} \otimes Q_t X^{x_t} |0\rangle \langle 0| X^{x_t} Q'_t \otimes \\
& \quad \sum_{P \in \mathbb{P}_n} Z^{g(P, c_i, d_i, x_i)} X^{f(P, c_i, d_i, x_i)} P R P X^{f(P, c_i, d_i, x_i)} Z^{g(P, c_i, d_i, x_i)} C_t E^{\delta_t} T_t \dots E^{\delta_{Q_2}} T_2 C_1 E^{\delta_{Q_1}} T_1 C_0 |0^n\rangle \\
& \quad \left. \langle 0^n | C_0^* T_1^* E^{\delta_{Q'_1}} C_1^* T_2^* E^{\delta_{Q'_2}} \dots T_t^* E^{\delta_{Q'_t}} C_t^* Z^{g(P, c_i, d_i, x_i)} X^{f(P, c_i, d_i, x_i)} P R' P X^{f(P, c_i, d_i, x_i)} Z^{g(P, c_i, d_i, x_i)} \right), \quad (42)
\end{aligned}$$

Where $f(P, c_i, d_i, x_i), g(P, c_i, d_i, x_i)$ are functions that depend on $P \in \mathbb{P}_n$, as well as $c_1, c_2, \dots, c_t, d_1, d_2, \dots, d_t$ and x_1, x_2, \dots, x_t , as given by the key update rules in Sections 4.1–4.4. We note that by these key update rules, it is a property of the protocol that the P 's appear uniformly at random for any choice of computation C .

We thus apply the same techniques as in Section 7.5.2 in order to simplify the above equation. First, note that we can use the Pauli twirl (Lemma 1) on the last n -qubit register; this yields:

$$\begin{aligned}
& \sum_{P_1, Q_1, Q'_1, P'_1, \dots, P_t, Q_t, Q'_t, P'_t, R} \alpha_{P_1} \alpha_{Q_1} \alpha_{P'_1}^* \alpha_{Q'_1}^* \dots \alpha_{P_t} \alpha_{Q_t} \alpha_{P'_t}^* \alpha_{Q'_t}^* |\alpha_R|^2 \\
& \quad \left(\sum_{c_1, d_1, e_1, x_1 \in \{0,1\}} X^{d_1} P_1 X^{d_1} |c_1\rangle \langle c_1| X^{d_1} P'_1 X^{d_1} \otimes Q_1 X^{x_1} |0\rangle \langle 0| X^{x_1} Q'_1 \otimes \dots \otimes \right. \\
& \quad \sum_{c_t, d_t, e_t, x_t \in \{0,1\}} X^{d_t} P_t X^{d_t} |c_t\rangle \langle c_t| X^{d_t} P'_t X^{d_t} \otimes Q_t X^{x_t} |0\rangle \langle 0| X^{x_t} Q'_t \otimes \\
& \quad \left. R C_t E^{\delta_t} T_t \dots E^{\delta_{Q_2}} T_2 C_1 E^{\delta_{Q_1}} T_1 C_0 |0^n\rangle \langle 0^n | C_0^* T_1^* E^{\delta_{Q'_1}} C_1^* T_2^* E^{\delta_{Q'_2}} \dots T_t^* E^{\delta_{Q'_t}} C_t^* R \right), \quad (43)
\end{aligned}$$

Next, we apply the Pauli twirl on the first $2t$ registers (assuming that the traced-out registers are decrypted); the result is

$$\begin{aligned}
& \sum_{P_1, Q_1, \dots, P_t, Q_t, R} |\alpha_{P_1}|^2 |\alpha_{Q_1}|^2 \dots |\alpha_{P_t}|^2 |\alpha_{Q_t}|^2 |\alpha_R|^2 \\
& \quad \left(\sum_{c_1, d_1, e_1, x_1 \in \{0,1\}} P_1 |c_1\rangle \langle c_1| P_1 \otimes Q_1 |0\rangle \langle 0| Q_1 \otimes \dots \otimes \sum_{c_t, d_t, e_t, x_t \in \{0,1\}} P_t |c_t\rangle \langle c_t| P_t \otimes Q_t |0\rangle \langle 0| Q_t \otimes \right. \\
& \quad \left. R C_t E^{\delta_t} T_t \dots E^{\delta_{Q_2}} T_2 C_1 E^{\delta_{Q_1}} T_1 C_0 |0^n\rangle \langle 0^n | C_0^* T_1^* E^{\delta_{Q'_1}} C_1^* T_2^* E^{\delta_{Q'_2}} \dots T_t^* E^{\delta_{Q'_t}} C_t^* R \right), \quad (44)
\end{aligned}$$

From the above, it is clear that benign attacks ($Q \in B_{t,n}$) lead to an acceptance probability of at most p , since in the case that Q is benign, each of the δ 's are 0, and so the correct computation, $C = C_t T \dots T C_1 T C_0$ is applied before the output qubit is measured (and furthermore, since Q is benign this output qubit is affected by either I or Z). However, for non-benign attacks, we bound their acceptance probability by the trivial bound of 1. This completes the proof. \square

7.6 Proof of Soundness

In order to complete the proof of soundness, we consider each possible fixed Pauli attack (the attacks are broken up into the same convex combination of Pauli attacks in all types of runs). Suppose that the input corresponds to a *no* instance. Consider the following cases.

1. ***P* is benign.** Then by Lemma 2, *V* accepts both test runs. In the computation run, by Lemma 4, *V* accepts with probability $\leq 1/3$ (as this is the probability that the honest computation outputs 0). The acceptance probability is thus at most $\frac{2}{3} + \frac{1}{9} = \frac{7}{9}$.
2. ***P* is not benign.** Then by Lemma 2, at least one of the *X* or *Z* test run rejects. The acceptance probability is thus at most $\frac{2}{3}$.

This completes the proof of Theorem 1. It is interesting to note that an optimal strategy to increase the acceptance probability is to behave honestly!

Acknowledgements

I am grateful to Urmila Mahadev for suggesting the relabelling described in Section 6.1, which simplifies the proof of soundness, and also for related discussions. It is a pleasure to thank Gus Gutoski for many deep conversations, from which this work originated, and Thomas Vidick for many related discussions. Furthermore, it is a pleasure to thank Jacob Krich for supplying background material on quantum simulations. I am also grateful to Harry Buhrman and Evelyn Wainwright for feedback on an earlier version of this work. A.B. is supported by Canada's NSERC and the University of Ottawa's Research Chairs program.

References

- [AA11] S. Aaronson and A. Arkhipov. The computational complexity of linear optics. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2011.
- [AA14] S. Aaronson and A. Arkhipov. Bosonsampling is far from uniform. *Quantum Information & Computation*, 14:1383–1423, 2014.
- [ABE10] D. Aharonov, M. Ben-Or, and E. Eban. Interactive proofs for quantum computations. In *Proceeding of Innovations in Computer Science 2010 (ICS 2010)*, pages 453–469, 2010.
- [AJL06] D. Aharonov, V. Jones, and Z. Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 427–436, 2006.
- [AMTW00] A. Ambainis, M. Mosca, A. Tapp, and R. De Wolf. Private quantum channels. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)*, pages 547–553, 2000.
- [AV14] D. Aharonov and U. Vazirani. Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics. In *Computability: Turing, Godel, Church and Beyond*. MIT press, 2014.

- [BCC88] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, October 1988. Preliminary version by Brassard and Crépeau in 27th FOCS, 1986.
- [BCG⁺02] H. Barnum, C. Crépeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. In *Proceedings of the 43rd Annual IEEE Symposium on the Foundations of Computer Science (FOCS '02)*, pages 449–458, 2002.
- [BFK09] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 517–526, 2009.
- [BFKW13] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther. Experimental verification of quantum computation. *Nature Physics*, 9:727–731, 2013.
- [BGS13] A. Broadbent, G. Gutoski, and D. Stebila. Quantum one-time programs. In *Advances in Cryptology CRYPTO 2013*, pages 344–360, 2013.
- [BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T -gate complexity. In *Advances in Cryptology–CRYPTO 2015*, pages 609–629, 2015.
- [BMP⁺00] P. Oscar Boykin, Tal Mor, Matthew Pulver, Vwani Roychowdhury, and Farrokh Vatan. On universal and fault-tolerant quantum computing. *Information Processing Letters*, 75:101–107, 2000.
- [BOCG⁺06] M. Ben-Or, C. Crépeau, D. Gottesman, A. Hassidim, and A. Smith. Secure multi-party quantum computation with (only) a strict honest majority. In *Proc. 47th IEEE Symposium on the Foundations of Computer Science (FOCS 2006)*, pages 249–260, 2006.
- [Bro15] A. Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.
- [CL92] A. Condon and R. Ladner. Interactive proof systems with polynomially bounded strategies. In *Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual*, pages 282–294, Jun 1992.
- [CLN05] A. M. Childs, D. W. Leung, and M. A. Nielsen. Unified derivations of measurement-based schemes for quantum computation. *Phys. Rev. A*, 71:032318, 2005.
- [DCEL09] C. Dankert, R. Cleve, J. Emerson, and E. Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80:012304, 2009.
- [DFPR14] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner. Composable security of delegated quantum computation. In *Advances in Cryptology ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 406–425, 2014.
- [DFSS05] I. B. Damgård, S. Fehr, L. Salvail, and C. Schaffner. Cryptography in the bounded quantum-storage model. In *Proc. 46th Annual IEEE Symposium on the Foundations of Computer Science (FOCS 2005)*, pages 449–458, 2005.

- [FBS⁺14] K. Fisher, A. Broadbent, L. K. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. J. Resch. Quantum computing on encrypted data. *Nature Communications* 5, 3074 (2014). *arXiv:1309.2586 [quant-ph]*, 5:3074, 2014.
- [Fey82] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.
- [FH15] Joseph F. Fitzsimons and Michal Hajdušek. Post hoc verification of quantum computation. Available as: *arXiv:1512.04375*, 2015.
- [FK12] J. F. Fitzsimons and E. Kashefi. Unconditionally verifiable blind computation. Available as: *arXiv:1203.5217v3*, 2012.
- [GC99] D. Gottesman and I. L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402:390–393, 1999.
- [GH05] Zhengting Gan and R. J. Harrison. Calibrating quantum chemistry: A multi-teraflop, parallel-vector, full-configuration interaction program for the Cray-X1. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 22–22, Nov 2005.
- [GKAE13] C. Gogolin, M. Kliesch, L. Aolita, and J. Eisert. Boson-sampling in the light of sample complexity. Available as: *arXiv:1306.3995*, 2013.
- [GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08*, pages 113–122, New York, NY, USA, 2008. ACM.
- [JJUW10] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. *Communications of the ACM*, 53(12):102–109, 2010.
- [KDK15] T. Kapourniotis, V. Dunjko, and E. Kashefi. On optimising quantum communication in verifiable quantum computing. In *Proceedings of the Asian Quantum Information Science Conference 2015 (AQIS 2015)*, pages 23–25, 2015.
- [KKMV08] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. In *Proceedings of the 2008 IEEE 23rd Annual Conference on Computational Complexity, CCC '08*, pages 211–222, Washington, DC, USA, 2008. IEEE Computer Society.
- [KW15] E. Kashefi and P. Wallden. Optimised resource construction for verifiable quantum computation. Available as: *arXiv:1510.07408*, 2015.
- [LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, October 1992.
- [NC00] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.
- [RUV13] B. W. Reichardt, F. Unger, and U. Vazirani. Classical command of quantum systems. *Nature*, 496:456–460, 2013.

- [Sha92] A. Shamir. $IP = PSPACE$. *J. ACM*, 39(4):869–877, October 1992.
- [SP00] P. W. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85:441–444, 2000.
- [SVB⁺14] N. Spagnolo, C. Vitelli, M. Bentivegna, D. J. Brod, A. Crespi, F. Flamini, S. Giacomini, G. Milani, R. Ramponi, P. Mataloni, R. Osellame, E. F. Galvão, and F. Sciarrino. Experimental validation of photonic boson sampling. *Nature Photonics*, 8(8):615–620, 2014.
- [Wat03] J. Watrous. $PSPACE$ has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575 – 588, 2003. Algorithms in Quantum Information Processing.
- [ZDC00] X. Zhou, D. Leung, and I. L. Chuang. Methodology for quantum logic gate construction. *Phys. Rev. A*, 62:052316, 2000.